Natural Language Processing

Parts-Of-Speech Tagging *and* Hidden Markov Models

BITS Pilani
Pilani Campus

innovate achieve lead

BITS Pilani

Pilani Campus

Session 6-7

These slides are prepared by the instructor, with grateful acknowledgement of Jurafsky and Martin and many others who made their course materials freely available online.

Session Content

POS Tagging and HMMs

- English Word Classes & Penn Treebank Tagset
- Part-of-Speech Tagging
- Markov Chains, Hidden Markov Model
- HMM Part-of-Speech Tagging

Inference in HMMs

- Likelihood Computation: The Forward Algorithm
- Decoding: The Viterbi Algorithm

Beyond HMMs

- Maximum Entropy Markov Model, Bidirectionality
- Neural Network Models for POS

This Session

Next Session

POS Tagging

Parts-Of-Speech Tagging

A word's grammatical role depends on how it is used in a sentence.

Knowing the Part-of-Speech (POS) of words helps NLP systems understand sentence structure and meaning and supports downstream tasks such as Parsing and Named-entity recognition.

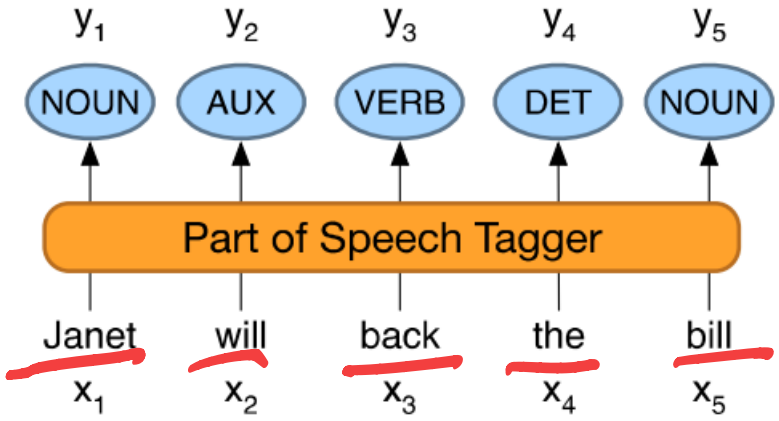
POS Tagging: Given a sequence of words:

$$w_1, w_2, \ldots, w_n$$

assign a POS tag to each:

$$t_1, t_2, \ldots, t_n$$

POS Tagging - Examples



The students are studying artificial intelligence

↓

The/DT students/NNS are/VBP studying/VBG artificial/JJ intelligence/NN

So, what is the big deal?

A word does not always have one POS.

- "The train leaves tomorrow." v/s "We train every day."
  train : noun / verb

- "They can fish." v/s "They ate fish."
  fish : verb / noun

POS-tagging has to perform this **disambiguation**.

Context to the rescue

The context resolves the ambiguity.

In the sentences: “I want to race.” and “I watched the race.”

- race could be: NN(noun) or VB(verb)

But:

- **“I want to race.”** strongly favours: race/VB

while:

- **“I watched the race.”** favours: race/NN

An effective POS tagging model needs to use context systematically.

English Word Classes

Word class	Role	Example
Noun	entity/concept	*dog, student*
Verb	action/state	*run, think*
Adjective	modifies noun	*large, red*
Adverb	modifies verb/adjective/etc.	*quickly*
Pronoun	stands for noun phrase	*he, they*
Determiner	specifies noun	*the, a*
Preposition	relation	*in, on*
Conjunction	connects	*and, but*
Auxiliary / Modal	grammatical function	*is, can*
…	…	
…	…	

Word Classes to POS Tagsets

Words can be described using broad linguistic categories:

- noun | verb | adjective | adverb | …

But an NLP system needs a **fixed inventory of tags** that can be assigned consistently to words.

Different corpora use different sets of Tags:

Tagset	Example tags	Characteristic
Brown Corpus	NN, VB, JJ, RB ...	One of the earliest large English POS-tagged corpora; relatively fine-grained
Penn Treebank (PTB)	NN, NNS, NNP, VB, VBD, VBG ...	Widely used in NLP; 36-tag tagset
Universal POS (UPOS)	NOUN, VERB, ADJ, ADV, DET ...	17 universal categories, designed for cross-linguistic consistency

Penn Treebank POS Tagset

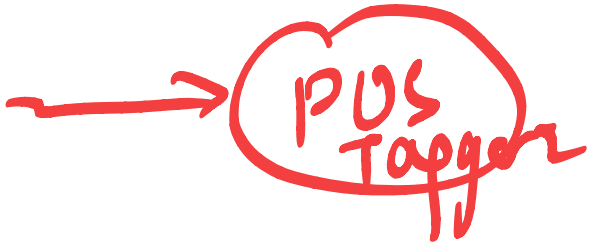
Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coord. conj.	*and, but, or*	NNP	proper noun, sing.	*IBM*	TO	infinitive to	*to*
CD	cardinal number	*one, two*	NNPS	proper noun, plu.	*Carolinas*	UH	interjection	*ah, oops*
DT	determiner	*a, the*	NNS	noun, plural	*llamas*	VB	verb base	*eat*
EX	existential 'there'	*there*	PDT	predeterminer	*all, both*	VBD	verb past tense	*ate*
FW	foreign word	*mea culpa*	POS	possessive ending	*'s*	VBG	verb gerund	*eating*
IN	preposition/ subordin-conj	*of, in, by*	PRP	personal pronoun	*I, you, he*	VBN	verb past participle	*eaten*
JJ	adjective	*yellow*	PRP$	possess. pronoun	*your*	VBP	verb non-3sg-pr	*eat*
JJR	comparative adj	*bigger*	RB	adverb	*quickly*	VBZ	verb 3sg pres	*eats*
JJS	superlative adj	*wildest*	RBR	comparative adv	*faster*	WDT	wh-determ.	*which, that*
LS	list item marker	*1, 2, One*	RBS	superlatv. adv	*fastest*	WP	wh-pronoun	*what, who*
MD	modal	*can, should*	RP	particle	*up, off*	WP$	wh-possess.	*whose*
NN	sing or mass noun	*llama*	SYM	symbol	+, %, &	WRB	wh-adverb	*how, where*

https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

POS tagging with the Penn Treebank Tagset

Tag the sentence:

- Janet will back the bill



Word	Tag	
Janet	NNP	*proper noun*
will	MD	*modal*
back	VB	*verb*
the	DT	*determiner*
bill	NN	*noun*

~~NN~~ | **VB** | ~~JJ~~ disambiguated by context

POS tagging as Sequence Labelling

Given a sequence of words:

$$W = w_1, w_2, \ldots, w_n$$

We want to assign a corresponding sequence of tags:

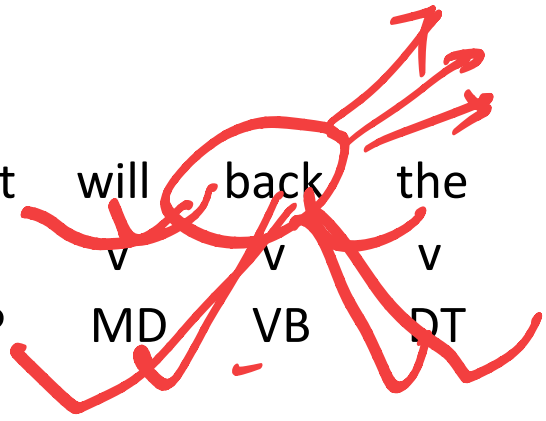
$$T = t_1, t_2, \ldots, t_n$$

Using Evidence:

- **Lexical** evidence**:** How compatible is this word with a particular tag?
- **Contextual** evidence**:** How likely is this tag given the surrounding tags?

Example :

Words:	Janet	will	back	the	bill
	v	v	v	v	v
Tags:	NNP	MD	VB	DT	NN



POS tagging - Problem Formulation

Given words W, find:

$$\hat{T} = \arg\max_{T} P(T \mid W)$$

Where:

- W: observed words
- T: candidate tag sequence
- $P(T \mid W)$: probability of tags given words

POS tagging - Problem Formulation

Using Bayes' rule:

$$P(T \mid W) = \frac{P(W \mid T)P(T)}{P(W)}$$

For a given sentence, $P(W)$ is constant.

Therefore, the optimal tag sequence is given by:

$$\hat{T} = \arg\max_{T} P(W \mid T)P(T)$$

Lexical Evidence
(how pausible is the word for this tag)

Contextual Evidence
(how pausible is the tag in given context)

innovate achieve lead

BITS Pilani

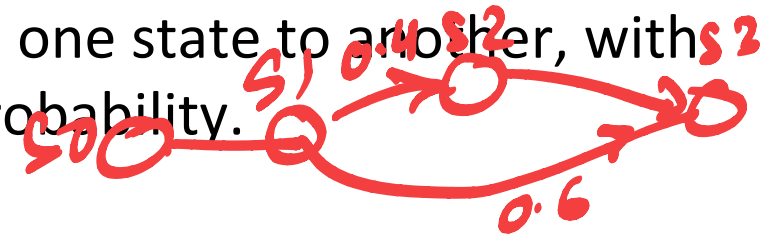
Pilani Campus

Markov Chains and HMM

The General Idea

Markov Chains

- A **Markov chain** is a model for a sequence of random variables in which the system moves from one state to another, with each transition governed by a probability.



- The key simplifying idea is the **Markov Assumption**: To predict the next state, we only need to know the current state; the earlier history does not provide any additional information.

- Instead of considering:

$$P(q_i \mid q_1, q_2, \ldots, q_{i-1})$$

- we only need:

$$P(q_i \mid q_{i-1})$$

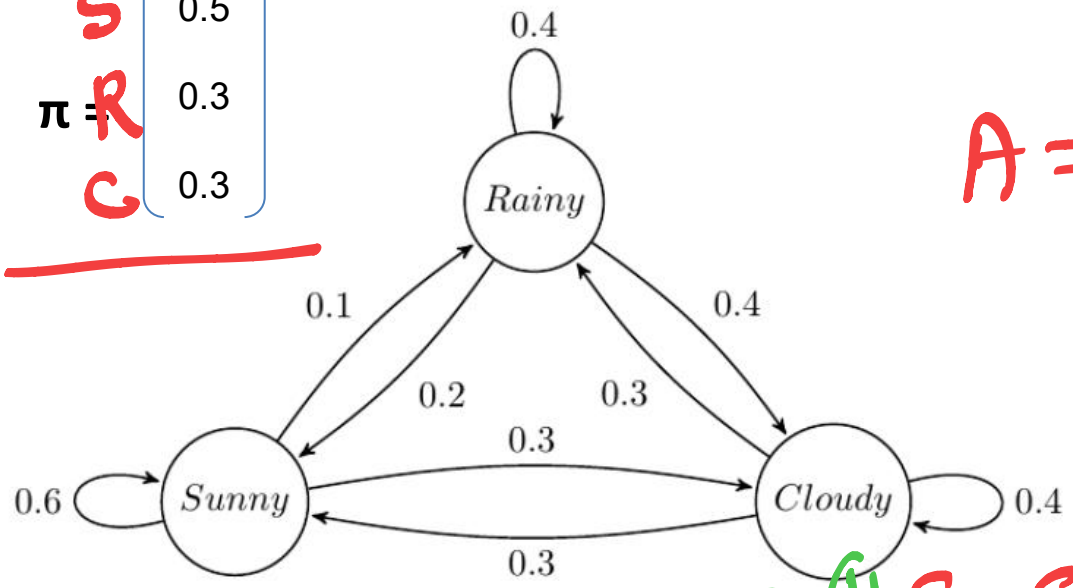
Markov Chains - Formal Definition

Formally, a Markov Chain is specified by the following components:

Q = $q_1\, q_2 \ldots q_N$	A set of N states
A = $a_{11}a_{12}\ldots a_{N1}\ldots a_{NN}$	A transition probability matrix **A**, each a_{ij} representing the probability of moving from state i to state j,
$\boldsymbol{\pi} = \pi_1,\pi_2,\pi_3,\ldots,\pi_N$	An initial probability distribution over states. $\boldsymbol{\pi}_i$ is the probability that the Markov chain will start in state i. *[Some states j may have $\boldsymbol{\pi}_j = 0$, meaning that they cannot be initial states.]*

Markov Chains - An Example

$\pi = \begin{matrix} S \\ R \\ C \end{matrix} \begin{bmatrix} 0.5 \\ 0.3 \\ 0.3 \end{bmatrix}$



$$P(q_1, \ldots, q_T) = \pi_{q_1} \prod_{i=2}^{T} P(q_i | q_{i-1})$$

$Q = \{S, R, C\}$

$$A = \begin{matrix} \\ S \\ R \\ C \end{matrix} \begin{matrix} \begin{matrix} S & R & C \end{matrix} \\ \begin{bmatrix} 0.6 & 0.1 & 0.3 \\ 0.1 & 0.4 & 0.4 \\ \cdots & \cdots & \cdots \end{bmatrix} \end{matrix}$$

$P(S \to S \to C \to R \to S'') = \pi(S) \cdot P(S|S) \cdot P(C|S) \cdot P(R|C) \cdot P(S|R)$

$= 0.5 \times 0.6 \times 0.3 \times 0.3 \times 0.2$

$= 0.0054$

Hidden Markov Models

- A **Markov chain** models a sequence of states that we can observe.

- But often, the states themselves are not directly observable.
  - Instead, we observe something that is probabilistically generated (emitted) by each state.

S(Machine) = Healthy, Degrading, Stalled

- A **Hidden Markov Model (HMM)** models:

  Hidden states ⟶ Observed events

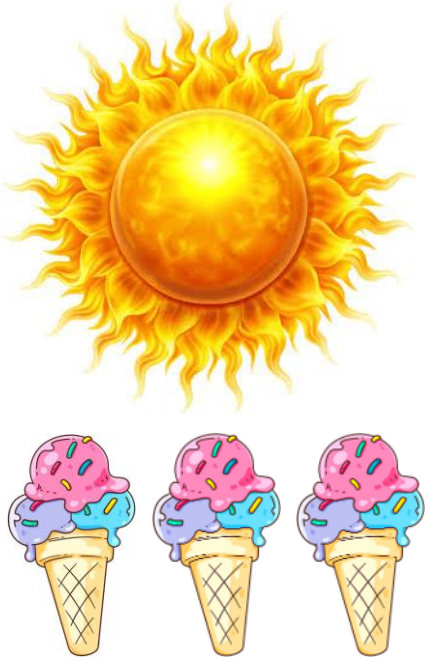
Hidden Markov Models - Formal Definition

Formally, a HMM is specified by the following components:

Q = $q_1 q_2 \ldots q_N$	A set of N states
A = $a_{11}a_{12}\ldots a_{N1}\ldots a_{NN}$	A transition probability matrix **A**, each a_{ij} representing the probability of moving from state i to state j,
B = $b_i(o_t)$	a sequence of observation likelihoods (or emission probabilities), each expressing the probability of an observation o_t (drawn from avocabulary V = $v_1, v_2, \ldots, v_V$) being generated from a state q_i
$\boldsymbol{\pi} = \pi_1, \pi_2, \pi_3, \ldots, \pi_N$	An initial probability distribution over states. $\boldsymbol{\pi}_i$ is the probability that the Markov chain will start in state i. *[Some states j may have $\boldsymbol{\pi}_j = 0$, meaning that they cannot be initial states.]*

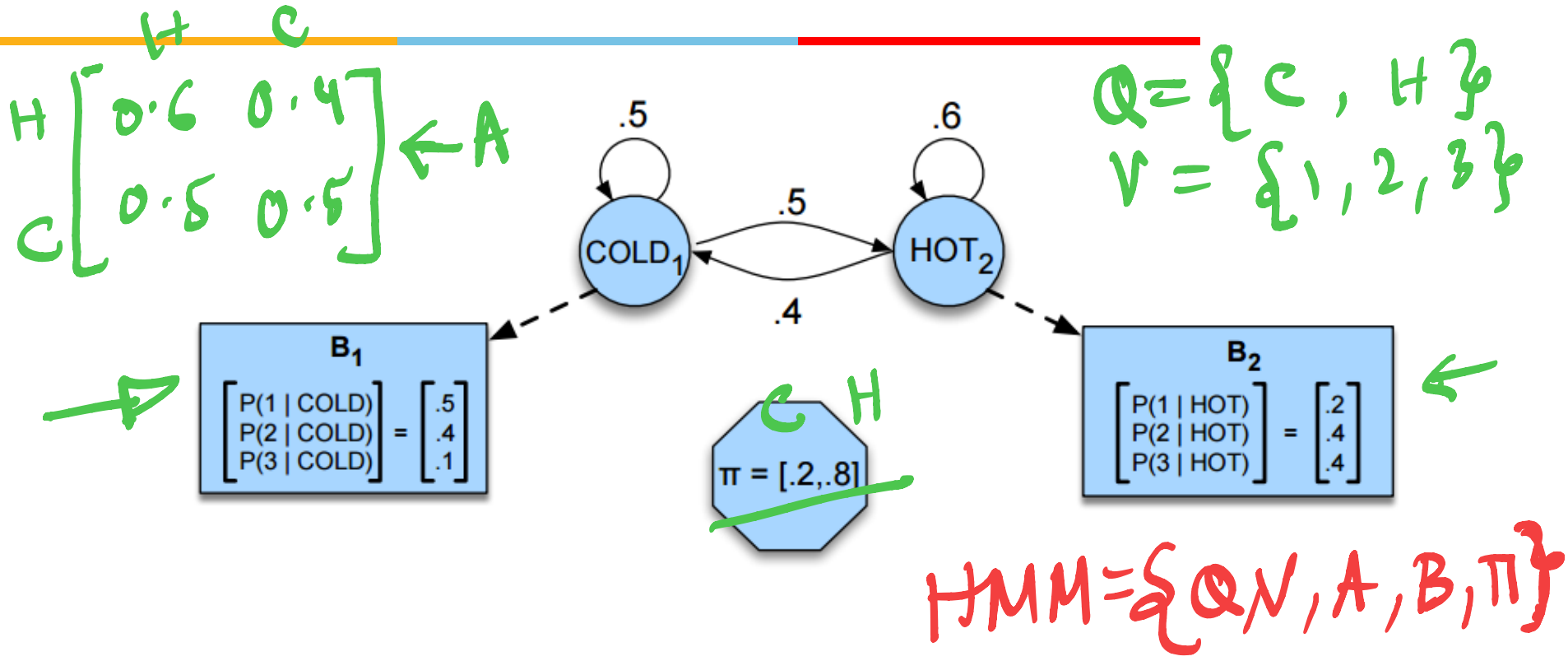
Hidden Markov Models - Example

- **HMM for Ice-Cream**:
  - Assume you are a climatologist in the year 2799 studying global warming.
  - You can't find any records of the weather in Baltimore for summer of 2007.
  - But you find Jason Eisner's diary which lists how many ice-creams Jason ate every day that summer



- **Eisner Task**:
  - *Given:*
    - Ice Cream Observation Sequence: 1,2,3,2,2,2,3…
  - *Produce:*
    - Hidden Weather Sequence: H,C,H,H,H,C,C…

Eisner Task HMM



Q:

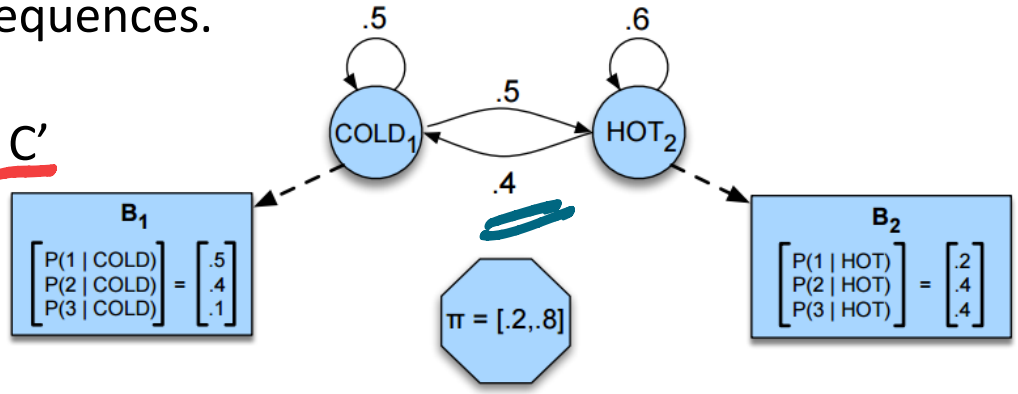
- Let's say we observed a sequence '1, 3, 1'
- What is the best potential weather sequence that could have caused this observaed sequence?

Eisner Task HMM

There are 8 possible weather sequences.

Let's consider just 1. Say, 'C, H, C'

- Q = C, H, C
- O = 1, 3, 1



Cold as the initial state
P(Cold|Start)

Observing a 1 on a cold day
P(1 | Cold)

Hot as the next state
P(Hot | Cold)

Observing a 3 on a hot day
P(3 | Hot)

Cold as the next state
P(Cold|Hot)

Observing a 1 on a cold day
P(1 | Cold)

$$P(Q|O) = P(C|START)\,P(1|C)\,P(H|C)\,P(3|H)\,P(C|H)\,P(1|C)$$
$$= 0.2 \times 0.5 \times 0.4 \times 0.4 \times 0.3 \times 0.5$$
$$= 0.0024$$

Best Sequence = $\arg\max_{Q} P(Q|O)$

HMMs - Three Key Problems

(Rabiner, 1989)

Given this framework there are 3 problems that we can pose to an HMM:

1. **Likelihood**: Given an HMM $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ and an observation sequence O, determine the likelihood $P(O|\lambda)$.

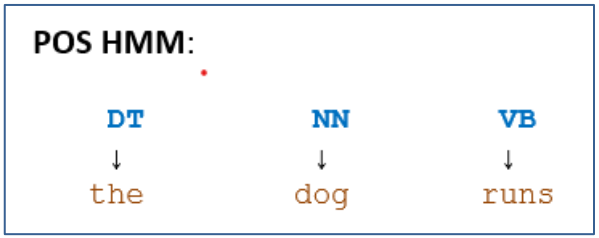
2. **Decoding**: Given an observation sequence O and a model $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, discover the most likely state sequence.

3. **Learning**: Given an observation sequence and a set of states in the HMM, find the best model parameters.

HMM for POS Tagging

We can model **POS tagging** as a **HMM inference problem**

- tags -> hidden states
- words -> observations

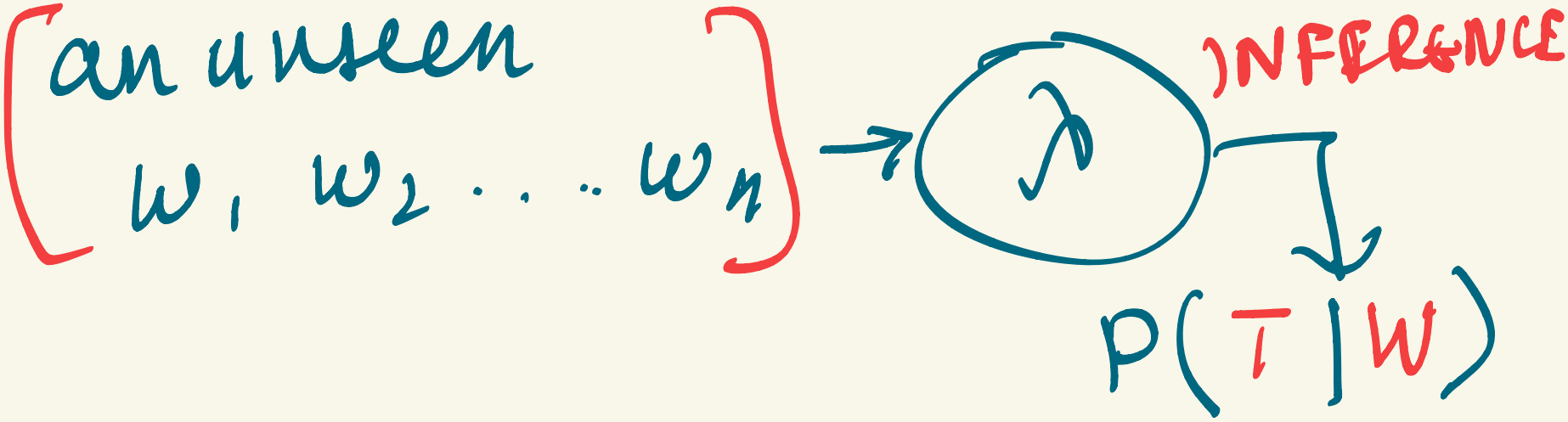


HMM	POS Tagging
Hidden state q_i	POS tag t_i
Observation o_i	Word w_i
State transition	Tag transition
$P(q_i \mid q_{i-1})$	$P(t_i \mid t_{i-1})$
Emission probability	$P(w_i \mid t_i)$
Initial probability	$P(t_1)$

$T = \{$ all possible tags $\}$ 27

$V = \{$ all possible words $\}$

$\lambda = \{ \pi, A, B \}$



HMM for POS Tagging - Assumptions

1. Markov assumption, for tag sequence:

The current tag depends only on the previous tag, i.e.

$$P(t_i \mid t_1, t_2, \ldots, t_{i-1}) = P(t_i \mid t_{i-1})$$

2. Output independence, for word generation:

The current word depends only on its corresponding tag, i.e.

$$P(w_i \mid t_1, t_2, \ldots t_n, w_1, w_2, \ldots w_{i-1}) = P(w_i \mid t_i)$$

HMM for POS Tagging - P(W|T)

- Consider "**the dog barks**" , with proposed tags: **DT NN VB**

- The HMM assigns a joint probability to this word/tag sequence:

$$P(W,T) = P(DT) \times P(the \mid DT) \times P(NN \mid DT) \times P(dog \mid NN) \times P(VB \mid NN) \times P(barks \mid VB)$$

- Generalizing:

$$P(W,T) = P(t_1)P(w_1|t_1)\prod_{i=2}^{n} P(t_i|t_{i-1})P(w_i|t_i)$$

context ev lexical ev

HMM for POS Tagging - Learning

$\lambda = \{\pi, A, B\}$

Using a tagged corpus, one can **count transitions** and **word/tag associations**.

```
the/DT dog/NN barks/VB
the/DT cat/NN sleeps/VB
a/DT dog/NN runs/VB
. . .
```

1. Transition probabilities, e.g.

$$P(NN \mid DT) = \frac{C(DT \rightarrow NN)}{C(DT \rightarrow *)}$$

2. Emission probabilities, e.g.

$$P(dog \mid NN) = \frac{C(NN, dog)}{C(NN)} \qquad \frac{2}{3} = 0.66$$

3. Initial probabilities, e.g.

$$P(DT) = \frac{C(\text{sentence starts with DT})}{C(\text{sentences})} \qquad = \frac{3}{3} = 1$$

HMM for POS Tagging - Inference

- Once we learn $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{\pi}$, the HMM model is known.
- To tag a new sentence, we need to find the best sequence:

$$\hat{T} = \arg\max_{T} P(T \mid W)$$

25 tags
10 words
$(25)^{10}$

- For n_t possible tags and n_w words, there are $(n_t)^{nw}$ possible tag sequences.
- Given the exponential nature of the problem, is there a way to find its solution efficiently?
  - We'll start with this question in the next session

Inference in HMMs

The Forward and Viterbi Algorithms

TBD

- A
- B
- C

Beyond HMMs

TBD

- A
- B
- C

innovate achieve lead

BITS Pilani

Pilani Campus

Thank You